Web-Based Unsupervised Learning for Query Formulation in Question Answering

Yi-Chia Wang[1], Jian-Cheng Wu[2], Tyne Liang[1], and Jason S. Chang[2]

[1] Dep. of Computer and Information Science, National Chiao Tung University,
1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C.
rhyme.cis92g@nctu.edu.tw, tliang@cis.nctu.edu.tw
[2] Dep. of Computer Science, National Tsing Hua University,
101, Section 2 Kuang Fu Road, Hsinchu, Taiwan 300, R.O.C.
d928322@oz.nthu.edu.tw, jschang@cs.nthu.edu.tw

Abstract. Converting questions to effective queries is crucial to open-domain question answering systems. In this paper, we present a web-based unsupervised learning approach for transforming a given natural-language question to an effective query. The method involves querying a search engine for Web passages that contain the answer to the question, extracting patterns that characterize fine-grained classification for answers, and linking these patterns with n-grams in answer passages. Independent evaluation on a set of questions shows that the proposed approach outperforms a naive keyword-based approach in terms of mean reciprocal rank and human effort.

1 Introduction

An automated *question answering* (QA) system receives a user's natural-language question and returns exact answers by analyzing the question and consulting a large text collection [1, 2]. As Moldovan et al. [3] pointed out, over 60% of the QA errors can be attributed to ineffective question processing, including query formulation and query expansion.

A naive solution to query formulation is using the keywords in an input question as the query to a search engine. However, it is possible that the keywords may not appear in those answer passages which contain answers to the given question. For example, submitting the keywords in "*Who invented washing machine?*" to a search engine like Google may not lead to retrieval of answer passages like "*The inventor of the automatic washer was John Chamberlain.*" In fact, by expanding the keyword set ("*invented*", "*washing*", "*machine*") with *"inventor of,"* the query to a search engine is effective in retrieving such answer passages as the top-ranking pages. Hence, if we can learn how to associate a set of questions (e.g. (*"who invented …?*") with effective keywords or phrases (e.g. *"inventor of"*) which are likely to appear in answer passages, the search engine will have a better chance of retrieving pages containing the answer.

In this paper, we present a novel Web-based unsupervised learning approach to handling question analysis for QA systems. In our approach, training-data questions are first analyzed and classified into a set of fine-grained categories of question

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 519 – 529, 2005.

patterns. Then, the relationships between the question patterns and n-grams in answer passages are discovered by employing a word alignment technique. Finally, the best query transforms are derived by ranking the n-grams which are associated with a specific question pattern. At runtime, the keywords in a given question are extracted and the question is categorized. Then the keywords are expanded according the category of the question. The expanded query is the submitted to a search engine in order to bias the search engine to return passages that are more likely to contain answers to the question. Experimental results indicate the expanded query indeed outperforms the approach of directly using the keywords in the question.

2 Related Work

Recent work in Question Answering has attempted to convert the original input question into a query that is more likely to retrieve the answers. Hovy et al. [2] utilized WordNet hypernyms and synonyms to expand queries to increase recall. Hildebrandt et al. [4] looked up in a pre-compiled knowledge base and a dictionary to expand a definition question. However, blindly expanding a word using its synonyms or dictionary gloss may cause undesirable effects. Furthermore, it is difficult to determine which of many related word senses should be considered when expanding the query.

Radev et al. [5] proposed a probabilistic algorithm called *QASM* that learns the best query expansion from a natural language question. The query expansion takes the form of a series of operators, including INSERT, DELETE, REPLACE, etc., to paraphrase a factual question into the best search engine query by applying Expectation Maximization algorithm. On the other hand, Hermjakob et al. [6] described an experiment to observe and learn from human subjects who were given a question and asked to write queries which are most effective in retrieving the answer to the question. First, several randomly selected questions are given to users to "manually" generate effective queries that can bias Web search engines to return answers. The questions, queries, and search results are then examined to derive seven query reformulation techniques that can be used to produce queries similar to the ones issued by human subjects.

In a study closely related to our work, Agichtein et al. [7] presented *Tritus* system that automatically learns transforms of wh-phrases (e.g. expanding "*what is*" to "*refers to*") by using FAQ data. The wh-phrases are restricted to sequences of function word beginning with an interrogative, (i.e. who, what, when, where, why, and how). These wh-phrases tend to coarsely classify questions into a few types. *Tritus* uses heuristic rules and thresholds of term frequencies to learn transforms.

In contrast to previous work, we rely on a mathematical model trained on a set of questions and answers to learn how to transform the question into an effective query. Transformations are learned based on a more fine-grained question classification involving the interrogative and one or more content words.

3 Transforming Question to Query

The method is aimed at automatically learning of the best transforms that turn a given natural language question into an effective query by using the Web as corpus. To that

end, we first automatically obtain a collection of answer passages (*AP*s) as the training corpus from the Web by using a set of (*Q*, *A*) pairs. Then we identify the question pattern for each *Q* by using statistical and linguistic information. Here, a question pattern Q_p is defined as a question word plus one or two keywords that are related to the question word. Q_p represents the question intention and it can be treated as a preference indicative for fine-grained type of named entities. Finally, we decide the transforms *T*s for each Q_p by choosing those phrases in the *AP*s that are statistically associated with Q_p and adjacent to the answer *A*.

Table 1. An example of converting a question (*Q*) with its answer (*A*) to a *SE* query and retrieving answer passages (*AP*)

(Q, A)	***AP***
What is the capital of Pakistan? Answer:(*Islamabad*)	Bungalow For Rent in *Islamabad*, Capital Pakistan. Beautiful Big House For …
	Islamabad is the capital of Pakistan. Current time, …
$(k_1, k_2, \ldots, k_n, A)$	
capital, Pakistan, Islamabad	…the airport which serves Pakistan's capital *Islamabad*, …

3.1 Search the Web for Relevant Answer Passages

For training purpose, a large amount of question/answer passage pairs are mined from the Web by using a set of question/answer pairs as seeds.

More formally, we attempt to retrieve a set of (*Q*, *AP*) pairs on the Web for training purpose, where *Q* stands for a natural language question, and *AP* is a passage containing at least one keyword in *Q* and *A* (the answer to *Q*). The seed data (*Q*, *A*) pairs can be acquired from many sources, including trivia game Websites, TREC QA Track benchmarks, and files of Frequently Asked Questions (FAQ). The output of this training-data gathering process is a large collection of (*Q*, *AP*) pairs. We describe the procedure in details as follows:

1. For each (*Q*, *A*) pair, the keywords $k_1, k_2, \ldots, k_n$ are extracted from *Q* by removing stopwords.
2. Submit $(k_1, k_2, \ldots, k_n, A)$ as a query to a search engine *SE*.
3. Download the top *n* summaries returned by *SE*.
4. Separate sentences in the summaries, and remove HTML tags, URL, special character references (e.g., "<").
5. Retain only those sentences which contain *A* and some k_i.

Consider the example of gathering answer passages from the Web for the (*Q*, *A*) pair where *Q* = "*What is the capital of Pakistan?*" and *A* = "*Islamabad*." See Table 1 for the query submitted to a search engine and potential answer passages returned.

3.2 Question Analysis

This subsection describes the presented identification of the so-called "question pattern" which is critical in categorizing a given question and transforming the question into a query.

Formally, a "question pattern" for any question is defined as following form:

question-word head-word+

where "question-word" is one of the interrogatives (Who/What/Where/When/How) and the "head-word" represents the headwords in the subsequent chunks that tend to reflect the intended answer more precisely. If the first headword is a light verb, an additional headword is needed. For instance, "*who had hit*" is a reasonable question pattern for "*Who had a number one hit in 1984 with 'Hello'?*", while "*who had*" seems to be too coarse.

In order to determine the appropriate question pattern for each question, we examined and analyzed a set of questions which are part-of-speech (POS) tagged and phrase-chunked. With the help of a set of simple heuristic rules based on POS and chunk information, fine-grained classification of questions can be carried out effectively.

Question Pattern Extraction

After analyzing recurring patterns and regularity in quizzes on the Web, we designed a simple procedure to recognize question patterns. The procedure is based on a small set of prioritized rules.

The question word which is one of the wh-words ("*who*," "*what*," "*when*," "*where*," "*how*," or "*why*") tagged as determiner or adverbial question word. According to the result of POS tagging and phrase chunking, we further decide the main verb and the voice of the question. Then, we apply the following expanded rules to extract words to form question patterns:

Rule 1: *Question word in a chunk of length more than one (see Example (1) in Table 2).*
Qp = question word + headword in the same chunk

Rule 2: *Question word followed by a light verb and Noun Phrase(NP) or Prepositional Phrase(PP) chunk (Example (2)).*
Qp = question word + light verb +headword in the following NP or PP chunk

Rule 3: *Question word followed immediately by a verb (Example (3)).*
Qp = question word + headword in the following Verb Phrase(VP) or NP chunk

Rule 4: *Question word followed by a passive VP (Example (4)).*
Qp = Question word + "to be" + headword in the passive VP chunk

Rule 5: *Question word followed by the copulate "to be" and an NP (Example (5)).*
Qp = Question word + "to be" + headword in the next NP chunk

Rule 6: *If none of the above rules are applicable, the question pattern is the question word.*

By exploiting linguistic information of POS and chunks, we can easily form the question pattern. These heuristic rules are intuitive and easy to understand. Moreover, the fact that these patterns which tend to recur imply that they are general and it is easy to gather training data accordingly. These question patterns also indicate a preference for the answer to be classified with a fine-grained type of proper nouns. In

the next section, we describe how we exploit these patterns to learn the best question-to-query transforms.

Table 2. Example questions and question patterns (of words shown in bold)

(1)	**Which** female **singer** performed the first song on Top of the Pops?
(2)	**Who** in 1961 **made** the first space **flight**?
(3)	**Who painted "**The Laughing Cavalier"?
(4)	**What is** a group of geese **called**?
(5)	**What** is the second longest **river** in the world?

3.3 Learning Best Transforms

This section describes the procedure for learning transforms *T*s which convert the question pattern Q_p into bigrams in relevant *AP*s.

Word Alignment Across *Q* and *AP*

We use word alignment techniques developed for statistical machine translation to find out the association between question patterns in *Q* and bigrams in *AP*. The reason why we use bigrams in *AP*s instead of unigrams is that bigrams tend to have more unique meaning than single words and are more effective in retrieving relevant passages.

We use Competitive Linking Algorithm [8] to align a set of (*Q*, *AP*) pairs. The method involves preprocessing steps for each (*Q*, *AP*) pair so as to filter useless information:

1. Perform part-of-speech tagging on *Q* and *AP*.
2. Replace all instances of *A* with the tag <ANS> in *AP*s to indicate the location of the answers.
3. Identify the question pattern, Q_p and keywords which are *not* a named entity. We denote the question pattern and keywords as $q_1, q_2, ..., q_n$.
4. Convert *AP* into bigrams and eliminate bigrams with low term frequency (tf) or high document frequency (df). Bigrams composed of two function words are also removed, resulting in bigrams $a_1, a_2, ..., a_m$.

We then align *q*'s and *a*'s via Competitive Linking Algorithm (CLA) procedure as follows:

Input: A collection *C* of (*Q*; *A*) pairs, where $(Q; A) = (q_1 = Q_p, q_2, q_3, ..., q_n; a_1, a_2, ..., a_m)$
Output: Best alignment counterpart *a*'s for all *q*'s in *C*

1. For each pair of (*Q*; *A*) in *C* and for all q_i and a_j in each pair of C, calculate LLR(q_i, a_j), logarithmic likelihood ratio (LLR) between q_i and a_j, which reflects their statistical association.
2. Discard (*q*, *a*) pairs with a LLR value lower than a threshold.

3. For each pair of (Q; A) in C and for all q_i and a_j therein, carry out Steps 4-7:
4. Sort list of (q_i, a_j) in each pair of (Q ; A) by decreasing LLR value.
5. Go down the list and select a pair if it does not conflict with previous selection.
6. Stop when running out of pairs in the list.
7. Produce the list of aligned pairs for all Qs and APs.
8. Tally the counts of aligning (q, a).
9. Select top k bigrams, t_1, t_2, ..., t_k, for every question pattern or keyword q.

The LLR statistics is generally effective in distinguishing related terms from unrelated ones. However, if two terms occur frequently in questions, their alignment counterparts will also occur frequently, leading to erroneous alignment due to indirect association. CLA is designed to tackle the problem caused by indirect association. Therefore, if we only make use of the alignment counterpart of the question pattern, we can keep the question keywords in Q so as to reduce the errors caused by indirect association. For instance, the question "*How old was Bruce Lee when he died?*" Our goal is to learn the best transforms for the question pattern "*how old*." In other words, we want to find out what terms are associated with "*how old*" in the answer passages. However, if we consider the alignment counterparts of "*how old*" without considering those keyword like "*died*," we run the risk of getting "*died in*" or "*is dead*" rather than "*years old*" and "*age of*." If we have sufficient data for a specific question pattern like "*how long*," we will have more chances to obtain alignment counterparts that are effective terms for query expansion.

Distance Constraint and Proximity Ranks

In addition to the association strength implied with alignment counts and co-occurrence, the distance of the bigrams to the answer should also be considered. We observe that terms in the answer passages close to the answers intuitively tend to be useful in retrieving answers. Thus, we calculate the bigrams appearing in a window of three words appearing on both sides of the answers to provide additional constraints for query expansion.

Combing Alignment and Proximity Ranks

The selection of the best bigrams as the transforms for a specific question pattern is based on a combined rank of alignment count and proximity count. It takes the average of these two counts to re-rank bigrams. The average rank of a bigram b is

$$Rank_{avg}(b) = (Rank_{align}(b) + Rank_{prox}(b))/2,$$

where $Rank_{align}(b)$ is the rank of b's alignment count and $Rank_{prox}(b)$ is the rank of b's proximity count. The n top-ranking bigrams for a specific type of question will be chosen to transform the question pattern into query terms. For the question pattern "*how old*," the candidate bigrams with alignment ranks, co-occurring ranks, and average ranks are shown in Table 3.

Table 3. Average rank calculated from for the bigram counterparts of "*how old*"

Bigrams	Alignment Rank	Proximity Rank	Avg. Rank	Final Rank
age of	1	1	1	1
years old	2	2	2	2
ascend the	3	-	-	-
throne in	4	3	3.5	3
the youngest	3	-	-	-
…	…	…	…	…

3.4 Runtime Transformation of Questions

At runtime, a given question Q submitted by a user is converted into one or more keywords and a question pattern, which is subsequently expanded in to a sequence of query terms based on the transforms obtained at training.

We follow the common practice of keyword selection in formulating Q into a query:

- Function words are identified and discarded.
- Proper nouns that are capitalized or quoted are treated as a single search term with quotes.

Additionally, we expand the question patterns based on alignment and proximity considerations:

- The question pattern Q_p is identified according to the rules (in Section 3.2) and is expanded to be a disjunction (sequence of ORs) of Q_p's headword and n top-ranking bigrams (in section 3.3)
- The query will be a conjunction (sequence of ANDs) of expanded Q_p, proper names, and remaining keywords. Except for the expanded Q_p, all other proper names and keywords will be in the original order in the given question for the best results.

Table 4. An example of transformation from question into query

<table>
<tr><th colspan="3">Question</th></tr>
<tr><td colspan="3">How old was Bruce Lee when he died?</td></tr>
<tr><th>Question pattern</th><th>Proper noun</th><th>Keyword</th></tr>
<tr><td>how old</td><td rowspan="3">"Bruce Lee"</td><td rowspan="3">died</td></tr>
<tr><th>Transformation</th></tr>
<tr><td>age of, years old</td></tr>
<tr><th colspan="3">Expanded query</th></tr>
<tr><td colspan="3">Boolean query: ("old" OR "age of" OR "years old") AND "Bruce Lee" AND "died"</td></tr>
<tr><td colspan="3">Equivalent Google query: (old || "age of" || "years old") "Bruce Lee" died</td></tr>
</table>

For example, formulating a query for the question "*How old was Bruce Lee when he died?*" will result in a question pattern "*how old*." Because there is a proper noun "*Bruce Lee*" in the question and a remaining keyword "*died*," the query becomes "('*old*' OR '*age of*' OR '*years old*') AND '*Bruce Lee*' AND '*died.*'" Table 4 lists the query formulating for the example question.

4 Experiments and Evaluation

The proposed method is implemented by using the Web search engine, Google, as the underlying information retrieval system. The experimental results are also justified with assessing the effectiveness of question classification and query expansion.

We used a POS tagger and chunker to perform shallow parsing of the questions and answer passages. The tagger was developed using the Brown corpus and WordNet. The chunker is built from the shared CoNLL-2000 data provided by CoNLL-2000. The shared task CoNLL-2000 provides a set of training and test data for chunks. The chunker we used produces chunks with an average precision rate of about 94%.

4.1 Evaluation of Question Patterns

The 200 questions from TREC-8 QA Track provide an independent evaluation of how well the proposed method works for question pattern extraction works. We will also give an error analysis.

Table 5. Evaluation results of question pattern extraction

	Two "good" labels	At least one "good" label
Precision (%)	86	96

Table 6. The first five questions with question patterns and judgment

Question	Question pattern	Judgment
Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?	Who-author	good
What was the monetary value of the Nobel Peace Prize in 1989?	What value	good
What does the Peugeot company manufacture?	What do manufacture	good
How much did Mercury spend on advertising in 1993?	How much	good
What is the name of the managing director of Apricot Computer?	What name	bad

Two human judges both majoring in Foreign Languages were asked to assess the results of question pattern extraction and give a label to each extracted question pattern. A pattern will be judged as "good" if it clearly expresses the answer preference of the question; otherwise, it is tagged as "bad." The precision rate of extraction for these 200 questions is shown in Table 5. The second column indicates the precision rate when both of two judges agree that an extracted question pattern is "good." In addition, the third column indicates the rate of those question patterns that are found to be "good" by either judge. The results imply that the proposed pattern extraction rules are general, since they are effective even for questions independent of the training and development data. Table 6 shows evaluation results for "two 'good' labels" of the first five questions.

We summarize the reasons behind these bad patterns:

- Incorrect part-of-speech tagging and chunking
- Imperative questions such as "*Name the first private citizen to fly in space.*"
- Question patterns that are not specific enough

For instance, the system produces "*what name*" for "*What is the name of the chronic neurological autoimmune disease which …* ?", while the judges suggested that "*what disease*.". Indeed, some of the patterns extracted can be modified to meet the goal of being more fine-grained and indicative of a preference to a specific type of proper nouns or terminology.

4.2 Evaluation of Query Expansion

We implemented a prototype of the proposed method called *Atlas* (Automatic Transform Learning by Aligning Sentences of question and answer). To develop the system of *Atlas,* we gathered seed training data of questions and answers from a trivia game website, called QuizZone[1]. We collected the questions posted in June, 2004 on QuizZone and obtained 3,851 distinct question-answer pairs. We set aside the first 45 questions for testing and used the rest for training. For each question, we form a query with question keywords and the answer and submitted the query to Google to retrieve top 100 summaries as the answer passages. In all, we collected 95,926 answer passages.

At training time, we extracted a total of 338 distinct question patterns from 3,806 questions. We aligned these patterns and keywords with bigrams in the 95,926 answer passages, identified the locations of the answers, and obtained the bigrams appearing within a distance of 3 of the answers. At runtime, we use the top-ranking bigram to expand each question pattern. If no such bigrams are found, we use only the keyword in the question patterns. The expanded terms for question pattern are placed at the beginning of the query.

We submitted forty-five keyword queries and the same number of expanded queries generated by *Atlas* for the test questions to Google and obtained ten returned summaries for evaluation. For the evaluation, we use three indicators to measure the performance. The first indicator is the mean reciprocal rank (*MRR*) of the first relevant document (or summary) returned. If the *r*-th document (summary) returned is the one with the answer, then the reciprocal rank of the document (summary) is *1/r.*

[1] QuizZone (http://www.quiz-zone.co.uk)

The mean reciprocal rank is the average reciprocal rank of all test questions. The second indicator of effective query is the recall at *R* document retrieved (Recall at *R*). The last indicator measures the human effort (*HE*) in finding the answer. *HE* is defined as the least number of passages needed to be viewed for covering all the answers to be returned from the system.

The average length of these test questions is short. We believe the proposed question expansion scheme helps those short sentences, which tend to be less effective in retrieving answers. We evaluated the expanded queries against the same measures for summaries returned by simple keyword queries. Both batches of returned summaries for the forty-five questions were verified by two human judges.

As shown in Table 7, the MRR produced by keyword-based scheme is slightly lower than the one yielded by the presented query expansion scheme. Nevertheless, such improvement is encouraging by indicating the effectiveness of the proposed method.

Table 8 lists the comparisons in more details. It is found that our method is effective in bringing the answers to the top 1 and top 2 summaries as indicated by the high Recall of 0.8 at $R = 2$. In addition, Table 8 also shows that less user's efforts are needed by using our approach. That is, for each question, the average of summaries required to be viewed by human beings goes down from 2.7 to 2.3.

In the end, we found that those bigrams containing a content word and a function word turn out to be very effective. For instance, our method tends to favor transforms

Table 7. Evaluation results of MRR

Performances	MRR
GO (Direct keyword query for Google)	0.64
AT+GO (Atlas expanded query for Google)	0.69

Table 8. Evaluation Result of Recall at R and Human Effort

Rank	Rank count		Recall at *R*	
	GO	AT+GO	GO	AT+GO
1	25	26	0.56	0.58
2	6	10	0.69	0.80
3	5	3	0.80	0.87
4	0	1	0.80	0.89
5	1	1	0.82	0.91
6	2	0	0.87	0.91
7	1	0	0.89	0.91
8	2	0	0.93	0.91
9	0	1	0.93	0.93
10	0	0	0.93	0.93
No answers	3	3		
Human Effort	122	105		
# of questions	45	45		
HE per question	2.7	2.3		

such as "*who invented*" to bigrams such as "*invented by*," "*invent the*," and "*inventor of*." This contrasts to conventional wisdom of using a stoplist of mostly function words and excluding them from consideration in a query. Our experiment also shows a function word as part of a phrasal term seems to be very effective, for it indicate an implied relation with the answer.

5 Conclusion and Future Work

In this paper, we introduce a method for learning query transformations that improves the ability to retrieve passages with answers using the Web as corpus. The method involves question classification and query transformations using a learning-based approach. We also describe the experiment with over 3,000 questions indicates that satisfactory results were achieved. The experimental results show that the proposed method provides effective query expansion that potentially can lead to performance improvement for a question answering system.

A number of future directions present themselves. First, the patterns learned from answer passages acquired on the Web can be refined and clustered to derive a hierarchical classification of questions for more effective question classification. Second, different question patterns, like "*who wrote*" and "*which author*", should be treated as the same in order to cope with data sparseness and improve system performance. On the other hand, an interesting direction is the generating pattern transformations that contain the answer extraction patterns for different types of questions.

References

1. Ittycheriah, A., Franz, M., Zhu, W.-J., and Rathaparkhi, A. 2000. IBM's statistical question answering system. In Proceedings of the TREC-9 Question Answering Track, Gaithersburg, Maryland.
2. Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y. 2000. Question answering in Webclopedia. In Proceedings of the TREC-9 Question Answering Track, Gaithersburg, Maryland.
3. Moldovan D., Pasca M., Harabagiu S., & Surdeanu M. 2002. Performance Issues and error Analysis in an Open-Domain Question Answering System. In Proceedings of the 40th Annual Meeting of ACL, Philadelphia, Pennsylvania.
4. Hildebrandt, W., Katz, B., & Lin, J. 2004. Answering definition questions with multiple knowledge sources. In Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational.
5. Radev, D. R., Qi, H., Zheng, Z., Blair-Goldensohn, S., Fan, Z. Z. W., and Prager, J. M. 2001. Mining the web for answers to natural language questions. In Proceedings of the International Conference on Knowledge Management (CIKM-2001), Atlanta, Georgia.
6. Hermjakob, U., Echihabi, A., and Marcu, D. 2002. Natural Language Based Reformulation Resource and Web Exploitation for Question Answering. In Proceeding of TREC-2002, Gaithersburg, Maryland.
7. Agichtein, E., Lawrence, S., and Gravano, L. Learning to find answers to questions on the Web. 2003. In ACM Transactions on Internet Technology (TOIT), 4(2):129-162.
8. Melamed, I. D. 1997. A Word-to-Word Model of Translational Equivalence. In Proceedings of the 35st Annual Meeting of ACL, Madrid, Spain.
9. Yi-Chia Wang, Jian-Cheng Wu, Tyne Liang, and Jason S. Chang. 2004. Using the Web as Corpus for Un-supervised Learning in Question Answering, Proceedings of Rocling 2004, Taiwan.